Finding Node Location And Time Sensive In Wireless Sensor And Adhoc Networks

Mevundi Srinatha
Department of Computer Science and Engineering,
Intell Engineering College,
Anantapur, Andhra Pradesh,India

Dr. G. Prakash Babu
Computer Science and Engineering Department,.
Intell Engineering College, Affiliated to JNTU,
Anantapur, Andhra Pradesh,India.

Abstract— **Wireless sensor network consist of large number of sensor devices spread over a larger filed. A sensor is a small device which is having one or more sensors, processor, memory, a radio and actuator. Each sensor device will sense the data and transfers the collected data along the network. To transfer the data, each sensor node should be present in the network. For that finding location is important so as to send and receive the data from one to another node. Finding the location in the ad-hoc[1] network is too difficult. In ad-hoc and wireless sensor network, every node is moving from one location to another location, when node is moving from one location to another location during that juncture there may be a chance of loss of data in the wireless sensor[2] network. Finding the accurate location [3] in the ad-hoc network and the time frame to transfer the data is not an easy task. Let's say i is the one node and j is the another node in the network, then finding the distance between the node is (i,j) € G, where G is the Euclidean space. Distance can be found in the way of d(i,j) and can be defined as d(p(i),p(j)), in this both local and non-local nodes can be noticed.**

Keywords-Localization; localizability; wireless sensor network; time sense; distance between nodes; ad-hoc networkcomponent.

I. INTRODUCTION

The growth of wireless technology has enormously grounded the context-aware applications in which the localization is the most important factor. In the recent past, more of methods have been proposed for finding the node localization, in that main method is finding nodes manually, but this seems to be unsuitable for larger networks. So, special node called beacons node was introduced. In this literature, it's referred as zone lead. What is localization? It is a node which is under the cover of one zone lead, that zone lead will instruct the node to transfer the data. The major challenge is, finding the local and non-local nodes in the sensor network and deciding the time frame.

In this graph [4], Zone1 is represented by few nodes whereas, Zone2 has two nodes. Zone2 nodes are localizable nodes and Zone1 are non-localizable nodes. Zone1 local nodes are not local to Zone2 and same goes with zone2.

Theorem 1: To find localizability of the node in zone localization. If a zone have n nodes and if the zone region is r, if and only if n € r n is a local node to Zone1 and non-local node to Zone2 [5].

The main interpretation of this work are as follows: Motivated by a real deployed sensor network, analyze the limitations of existing works on or related to node localizability, scattered over different literatures, need of the hour is to define the localizability conditions, in this, the second issue is to detect how many local and non- local nodes are present in the sensor network. The third issue is, to find out whether local nodes are connected or not in the wireless sensor network.

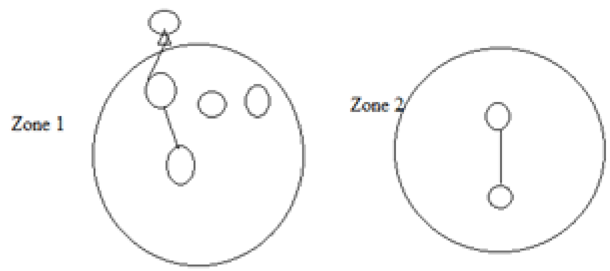

Figure 1. Sensor Network

Based on Theorem 1, global rigidity can be tested in polynomial time by combining existing algorithms for rigidity and 3-connectivity . If fixing any group of three vertices to avoid trivial variations in 2D plane (i.e., translation, rotation, and reflection), a globally rigid graph is uniquely realizable. Accordingly, a network with at least three beacons is localizable if and only if its distance graph is globally rigid. For node localizability, however, no such conclusion is presented so far. The rest of the paper is organized as follows: We discuss information about network in the session 2 and important conditions in the session 3 and 4. The prototype implementation we will discuss in 5. , and conclude the work in Section 6.

II. PRELIMINARY

In this section we will discuss about the region generation that region should be under the cover of network[6], without network we cannot develop this application, we can assume that where G is a distance graph in the functionality of p , G is a connected graph and having at least four vertices, in the Euclidean space.

Generally, realizations are referred to the feasible ones that respect the pair wise distance constraints between a pair of vertices i and j if the

Edge(i,j) € E . That is to say, d(p(i),p(j))=d(I,j) for all (i,j) € E. Two realization of G are equivalent if they are identical under translations, rotations, and reflections in 2D plane. A distance graph G has at least one feasible realization which represents the ground truth of the corresponding network. Formally, G is embeddable in 2D [7] space and all pair wise distances are compatible.

A graph is called generically rigid if one cannot continuously deform its realization while preserving distance constrains . the vertex in the graph are independent, all the nodes are realization space that are rigid

For a distance graph we are generating the nodes which are rigid in the space, these nodes are not unique in realization space , if it is region we can remove any node that means that node is crash and we can recover that node from the sensor network.

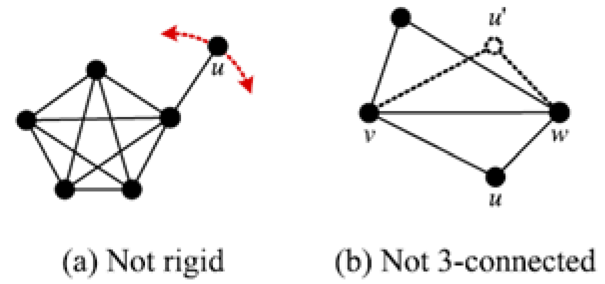

Figure 2. Realization nonuniqueness

That means we can remove the node u from the region of the sensor , that means we may be remove we may not be remove that node from the sensor, that is necessary and sufficient condition for globally rigid nodes

III. FIND NODE LOCALIZABILITY IN THE SENSOR NETWOTK

Based on the previous study of the network To find the node localizability it is very easy task , in this session we are showing that work in the form graph. To find the node localizability sensor is must then sensor will give the full information about that node, in this session we are giving the range of the node where it should be locate, depend on the range we can find that, it is local node or not

A. Creation of Connection Paths

We have observed that some network is required to develop this application, when network is created then we can create the paths of the nodes in the network With that network we can send the data from one node to another node, nodes can send the data in wireless network with path creation in the network, when network is created , the sensor have to check the nodes are local or non-local, when the nodes are local then it can send the data to any node in the local, the nodes may be available in the edge of the sensor that edge of the sensor that time that node may be transfer the data or may not be transfer the data to another node. When we sending the data from one node to another node time must be start, when data is start to send in that time, time is started and when the data is reached to destination in that time the time will stop automatically, and it will show in how much time we can send the data, in the sensor network we are finding the distance between the nodes In this graph it is showing the dowel arrow distance between the nodes and time sense.

IV. SUFFICIENT CONDITION FOR NODE AVAILABILITY

This sufficient condition based on the rigid of the nodes, these nodes are at a particular location in the sensor network, the location of the nodes are given in the below graph , every node have an rigid location , it is uniquely localizable. For convenience we denote this condition as RRT standing for Redundant Rigidity Try-connected. Note that a localizable vertex does not necessarily satisfy RRT, as shown in Fig. 4a. The graph consists of three beacon vertices (denoted by white circles) and three non-beacon vertices (denoted by black ones). It is clear that u is not in the 3-connected component of three beacon vertices.

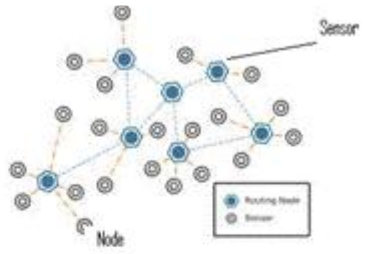

Figure 3. Routing in Sensor Network

However u's location [8] can be uniquely determined under the configuration. The possible reason is the distance between u and v is actually fixed although no edge connects them. If we add the edge (u,v) to G, u can be easily identified as localizable since the distances from u to 3 beacon vertices are available. This observation leads us to explore the implicit edges for identifying localizable vertices. In this graph we have floating sensor, that can transfer [9] data from one sensor to another sensor.

Let R denote the set of all realizations of G. For simplicity, let dr(u, v) instead of d(r(u), r(v)) denote the Euclidean distance between the two vertices u and v in a specific realization r €R.

Let DG(u, v)=Ur€R{ dr(u,v)}. For a rigid graph G;R is finite although |R| can be exponential to the size of G. As a result, DG(u,v) is finite since the number of distinct values of DG(u, v) is at most |R|.

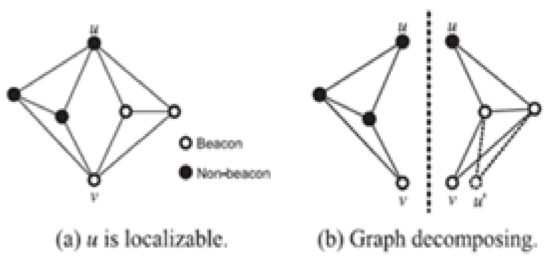

Figure 4. RRT Graph Representation

Definition(Inside Edge).In a distance graph gravity G=(V,E), an edge(u,v) is inside the region if (u,v)¥ E and in all realization of G, the distance between u and v always same in the region.

If(u,v) are implicit edge, it is equalent to the fact that DG(u,v) contains unique values, based on this concept of implicit edge, we define the file transfer time and distance between the nodes.

Definition 2 (Time sense and distance graph) .For a distance graph g=(V,E), its extended distance graph is GI=(V,E U EI) where EI is the set of implicit nodes. When we sending the file from one place to another place , starting time is t1, and after reaching the file the time is t2, we can find this time sensive in two ways one is depend on our system time and if it is in one system.

Startingtime-sendingtime=timesensive

i.e. t1-t2=timesensive(Te)

And second way is select the sending node system time, and receiver node system time, then calculate the time sensive . sending node can be represent as N1 and received node can be represent as N2.

Sendingnodetime-receivednodetime=timesensive

This can be represent as N1-N2=TimeSensive

```
if time t1=0
{
T1 Time i=0
∑ result= t1+t2
i<=T2 time
}
Return result;
Else
Return
Time is 0
```

A. *Finding Shortest Path*

We are finding the shortest path in the application, when we sending file from one place to another place, we sending through intermediate nodes. We have to find intermediate node in the sensor network. Through which we can send the data easily we have to select that node ,though we have to send the data to destination. The main aim of this we have to find the intermediate node and send the data to destination node in the less time, if any node is crashed in that time, we have to find alternate node in that time,

Finding node distance

N1 Distance is 10Miter from N2 and 12 Miter from N3 and 15 Miter from N3, then it will check for shortest distance is

```
If((N2< N3)&&( N2< N4)))
{
Data Transfer to   N2;
}
Else
{
Data Transfer to N5 ;
}
```

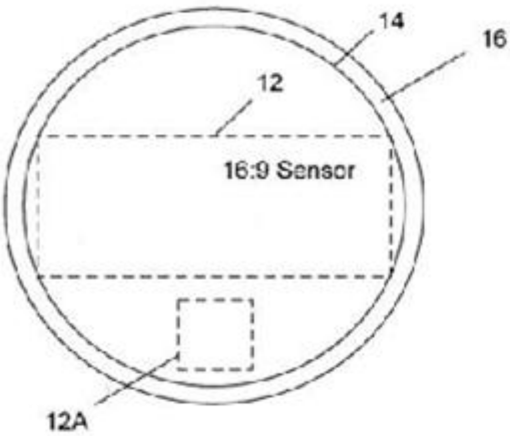

Figure 5. Sensor

In this we have to find the shortest path to the node, if shortest path is not found then we have to relevant path theory[10].

B. *Relevant Path*

In the relevant path theory we have to find the another path to send the data to destination node, if shortest path node having any problem. That is the use of the relevant path theory, in this graph , we have 5 sensors and nodes on that graph, in that graph we have the shortest path, through the shortest path we can transfer the data.

To find the shortest path we have to know the sensor information that location [11].

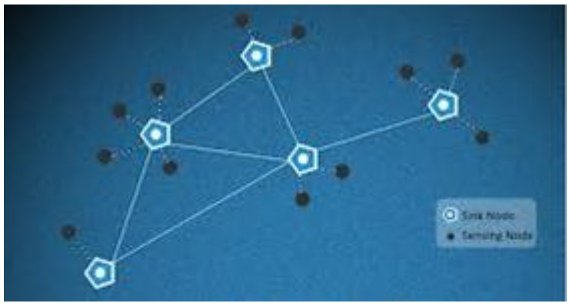

Figure 6. Sink Node in Wireless Sensor Network

C. *Extended Distance Graph*

Theorem 3: Let G1 denote the extended distance graph of G(V,E), which has a set B < V of k _>=3 vertices at known locations. If a vertex belongs to a globally rigid subgraph of G1 that contains at least three vertices in B, it is uniquely localizable in G.

D. *Sufficiency of RR3P Condition*

Theorem 3 provides so far the best sufficient condition for node localizability. However, it requires the knowledge of implicit paths connecting it to three beacon vertices. We call this condition RR3P for short. Note that RR3P is fundamentally different from the previously mentioned RR-3P. RR3P requires the three paths strictly residing in the redundantly rigid component to avoid the unexpected case in Fig. 4. We use the similar terms to show their close relationship.

Due to the necessity of redundant rigidity, for convenience, we assume G is redundantly rigid; otherwise let G denote the redundantly rigid component containing B. If G is 3-connected,

it is trivial that all vertices are localizable since G itself is globally rigid, so we focus on the only interesting case that G is not 3-connected. There exist two vertices v and w whose removal disconnects G. As a result, as shown in Fig. 7a, G can be divided into several overlapped and connected components Gi such that

$G = \bigcup G_i$ and $V(G_i \cap G_j)=\{v,w\}$ for all $i \neq j$

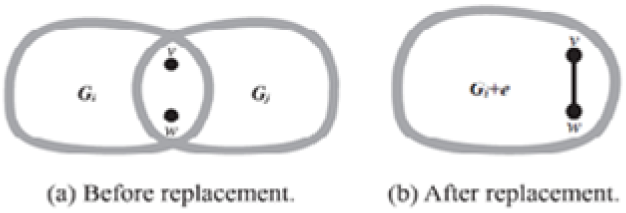

Figure 7. Edge Replacement Graph

For any specific Gi, we replace other components Gj(j≠i) by an edge=(v;w). This operation, as illustrated in Fig. 7,is defined as edge replacement.

V. PERFORMANCE EVALUATIONS

A. Experiment

In this experiment we are finding the accurate result of the node moving from one place to another place, when the data is transferred from one node to another node the result will be displayed accurately, The time stamp will start when a node starts sending the data and time stamp will stop once the data is received by the destination node. So, that one can determine the exact time frame which is required for transferring the data. We are doing this experiment on the 100 nodes.

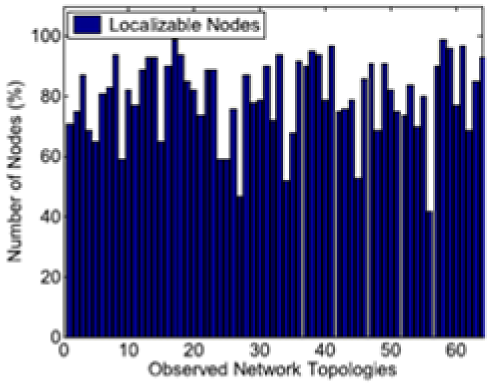

Figure 8. A Large Portion of Nodes are Localizable

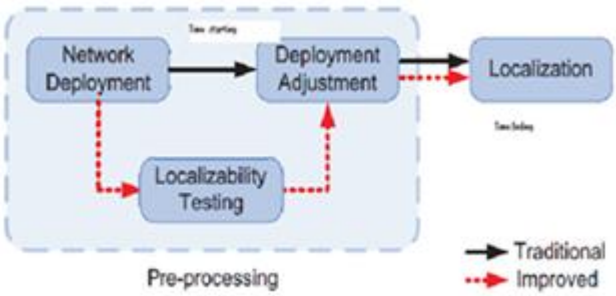

Figure 9. Localizability Assists Network Deployment

In this we can find that the 90 nodes are local , and other nodes are moving from the zones, those are moved zones are unable to send the data from one node to another node As shown in Fig. 9, the major difference of the improved flow is that the task of localizability testing is added to assist deployment adjustment. In detail, the testing algorithm is carried out on the initial network deployment and the results are used to instruct the subsequent adjustments. In the experiment, we enhance distance ranging capability through augmenting signal power. More specifically, we keep those localizable nodes unchanged while increase distance ranging of non-localizable ones from 5 to 25 percent.

VI. CONCLUSIONS

There is a limitation of network localizability. It is not practically possible that all the nodes in the network are localizable. The only solution for localizability is to send and receive the data from one node to another node. Here in this study we have referred the necessary and sufficient conditions for network localizability. Based on the research work, we can answer few fundamental questions on network localizability, they are: What are the nodes that are localizable or non-localizable.

In this study, we are finding the distance between the nodes, and finding the time frame within which the data is transferred from node to another node.

Further, one more important concept is time tracking. If the node is not responded with in time constraint, then we can say that it is non- localizable. If it has responded and it is under the cover of zone leader then we can conclude that it is a localizable node.

REFERENCES

[1] N.B. Priyantha, A. Chakraborty, and H. Balakrishnan, “The Cricket Location-Support System,” Proc. ACM MobiCom, 2000.

[2] S.Y. Seidel and T.S. Rappaport, “914 MHz Path Loss Prediction Models for Indoor Wireless Communications in Multifloored Buildings,” IEEE Trans. Antennas and Propagation, vol. 40, no. 2, pp. 209-217, Feb. 1992.

[3] T. Eren, D.K. Goldenberg, W. Whiteley, Y.R. Yang, A.S. Morse, B.D.O. Anderson, and P.N. Belhumeur, “Rigidity, Computation, and Randomization in Network Localization,” Proc. IEEE INFOCOM, 2004.

[4] B. Hendrickson, “Conditions for Unique Graph Realizations,” SIAM J. Computing, vol. 21, no. 1, pp. 65-84, 1992.

[5] B. Jackson and T. Jordan, “Connected Rigidity Matroids and Unique Realizations of Graphs,” J. Combinatorial Theory Series B, vol. 94, no. 1, pp. 1-29, 2005.

[6] D. Goldenberg, A. Krishnamurthy, W. Maness, Y.R. Yang, A. Young, A.S. Morse, A. Savvides, and B. Anderson, “Network Localization in Partially Localizable Networks,” Proc. IEEE INFOCOM, 2005.

[7] D.J. Jacobs and B. Hendrickson, “An Algorithm for Two-Dimensional Rigidity Percolation: The Pebble Game,” J. Computational Physics, vol. 137, pp. 346-365, 1997.

[8] J.E. Hopcroft and R.E. Tarjan, “Finding The Triconnected Components of a Graph,” Technical Report TR 140, Dept. of Computer Science, Cornell Univ., 1972.

[9] Z. Yang, M. Li, and Y. Liu, “Sea Depth Measurement with Restricted Floating Sensors,” Proc. IEEE 28th Int’l Real-Time Symp. (RTSS), 2007.

[10] A. Savvides, C. Han, and M.B. Strivastava, “Dynamic Fine- Grained Localization in Ad-Hoc Networks of Sensors,” Proc. ACM MobiCom, 2001.

[11] Z. Yang and Y. Liu, “Quality of Trilateration: Confidence Based Iterative Localization,” IEEE Trans. Parallel and Distributed Systems,